CLAIMS

1. A method for evaluating the locality of references for a layout of a computer program, the computer program to be executed by a computer system with a page architecture, each page having a multiplicity of memory locations, the method comprising:

for each of a plurality of memory locations of a page, calculating a metric value indicating a working set size of the layout when the layout is positioned to start at that memory location; and

combining the calculated metric values as an indication of the locality of references of the layout of the computer program.

2. The method of claim 1 wherein the plurality of memory locations include each memory location of a page.

3. The method of claim 1 wherein the plurality of memory locations include less than all the memory locations on a page.

4. The method of claim 1 wherein the plurality of memory locations are separated by relatively prime number of memory locations.

5. The method of claim 1 wherein the plurality of memory locations includes less than all the memory locations on a page and are approximately evenly distributed throughout the page.

6. The method of claim 1 wherein the plurality of memory locations includes less than all the memory locations on a page, are approximately evenly distributed throughout the page, and are separated by relatively prime number of memory locations.

7. The method of claim 1 wherein the combining of the calculated metric values includes averaging the calculated metric values.

8. A method in a computer system for determining a number of different layouts of a program image that should be evaluated when selecting a layout as a starting layout for a working set optimization process, the method comprising:

collecting data on results of using the working set optimization process to improve the working set of various layouts of various program images; and

determining from the collected number of layouts whose computational expense in generating and evaluating the layouts is worth an anticipated reduction in the working set when the program image is optimized by the working set optimization process.

9. The method of claim 8 wherein the collecting of data includes calculating a marginal density of the working set size of the optimized layout that is produced from one of the various layouts with the smallest locality of reference.

10. The method of claim 8 wherein the collecting of data includes calculating a mean marginal density that is

$$\mu = \int_{-\infty}^{\infty} y\, g(y)\, dy$$

where

$$g(y) = N \int_{-\infty}^{\infty} f(x,y) \left(\int_{-\infty}^{\infty} \int_{x}^{\infty} f(t,u)\, dt\, du \right)^{N-1} dx$$

where N is the number of layouts and f is a joint probability density function that relates a metric for each unoptimized layout to a metric for each corresponding optimized layout.

11. The method of claim 10 wherein the joint probability density function is

$$f(x,y) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(x^2-2\rho xy+y^2)}{2(1-\rho^2)}}$$

and where ρ is a normal correlation coefficient between a locality of reference metric for the unoptimized layouts and a working set size metric for the optimized layouts.

12. The method of claim 10 wherein the anticipated reduction is

$$\mu\sigma$$

where σ is a standard deviation of a working set size metric for the optimized layouts.

13. The method of claim 10 wherein the probability density function is a standard bivariate normal distribution.

14. The method of claim 8 wherein the determining of the number of layouts includes determining the computational expense of generating

and evaluating successively larger number of layouts until the computation expense is not worthwhile.

15. A method for estimating a rate of change per time period in results of a system, the system producing a result for each of a plurality of steps performed by the system, the steps occurring at irregular time intervals, the method comprising:

estimating the change in results at each of the plurality of steps;

estimating the time for each step; and

combining the estimated change in results per step with the estimated time per step to estimate the rate of change per time period in the results of the system.

16. The method of claim 15 wherein the estimating of the change in results includes filtering the actual change in results from one step to the next step.

17. The method of claim 15 wherein the estimating of the time for each step includes evaluating a formula that estimates the time per step based on estimated numbers of times that various sub-steps are performed per step.

18. The method of claim 17 wherein the estimated numbers of times that various sub-steps are performed are based on evaluating actual results of the system.

19. The method of claim 15 wherein the estimating of the change in results includes filtering the actual change in results from one step to the next step and wherein the estimating of the time for each step includes

evaluating a formula that estimates the time per step based on number of time various sub-steps are performed.

20. The method of claim 15 wherein the system is a computer program layout optimization system.

21. The method of claim 15 including terminating the system when the estimated rate of change per time period is outside of a threshold rate of change.

22. The method of claim 15 wherein the estimating of the change in the results includes filtering the actual change in results from one step to the next with a filter that is generated from analysis of results produced by the system.

23. The method of claim 22 wherein the filter is generated by using a frequency-domain analysis of an actual rate of change per step and defined rate of change per step, the defined rate of change being calculated for each step based on knowledge of results of subsequent steps.

24. The method of claim 22 wherein the filter is generated by using a time-domain analysis of an actual rate of change per step and defined rate of change per step, the defined rate of change being calculated for each step based on knowledge of results of subsequent steps.

25. A method in a computer system for estimating the rate of improvement in the working set for a plurality of incrementally improved layouts of a computer program, the method comprising:

estimating the change in working set size from one incrementally improved layout to the next incrementally improved layout;

estimating the time needed to incrementally improve the layout; and

combining the estimated change in working set size with the estimated time needed to incrementally improve the working set for that layout to estimate the rate of improvement.

26. The method of claim 25 wherein the estimating of the change in working set size includes filtering the actual change in working set size from one incremental improvement to the next.

27. The method of claim 25 wherein the estimating of the time to incrementally improve the layout includes evaluating a formula that estimates the time based on estimated numbers of times that various sub-steps are performed when incrementally improving the layout.

28. The method of claim 27 wherein the estimated numbers of times that various sub-steps are performed are based on evaluating actual results of the incremental improvement.

29. The method of claim 25 wherein the estimating of the change includes filtering the actual change in working set size from one incremental improvement to the next and wherein the estimating of the time to incrementally improve the layout includes evaluating a formula that estimates the time based on number of times various sub-steps are performed when incrementally improving the layout.

30. The method of claim 25 wherein the estimating of the change in working set size filters an actual change in working set size and the estimating of time computes the time based on number of times the layout has been incrementally improved.

31. The method of claim 25 including terminating the system when the estimated rate of change per time period is outside of a threshold rate of change.

32. The method of claim 25 wherein the estimating of the change in the working set size includes filtering the actual change in working set size from one incremental improvement to the next with a filter that is generated from analysis of working set size produced during the incremental improvement of other layouts.

33. The method of claim 32 wherein the filter is generated by using a frequency-domain analysis of an actual rate of change per incremental improvement and defined rate of change per incremental improvement, the defined rate of change being calculated for each incremental improvement based on knowledge of working set size of subsequent incremental improvements.

34. The method of claim 32 wherein the filter is generated by using a time-domain analysis of an actual rate of change per incremental improvement and defined rate of change per incremental improvement, the defined rate of change being calculated for each incremental improvement based on knowledge of working set size of subsequent incremental improvements.

35. A method in a computer system for estimating a rate of change of results of a function over an indeterminate interval, the method comprising:

receiving a plurality of results of the function;

calculating a negative slope of a convex hull of the results of the function; and

using the calculated negative slope as an estimate of the rate of change of results of the function.

36. The method of claim 35 wherein the results of the function are generated by calculating a working set size of a program image as the layout is incrementally improved.

37. The method of claim 35 including calculating a running minimum of the results of the function wherein the calculating of the negative slope of a convex hull does so for the calculated running minimum.

38. The method of claim 37 wherein the results of the function are generated by calculating a working set size of a program image as the layout is incrementally improved.

39. A method in a computer system for identifying coefficients for a filter, the filter for filtering results of a function, the method comprising:

collecting sample input values to the filter;

identifying desired output values from the filter for the collected sample input values;

generating a power spectrum of the collected sample input values;

generating a power spectrum of the identified desired output values;

calculating a difference between the generated power spectra; and

identifying coefficients that minimize differences between the calculated differences.

40. The method of claim 39 wherein the calculated difference is offset so that the calculated difference at a frequency of zero is zero.

41. The method of claim 39 wherein the results of the function are generated by calculating a working set size of a program image as the working set size is incrementally improved.

42. The method of claim 39 wherein a Levenberg-Marquardt algorithm is used to identify the coefficients.

43. A method in a computer system for identifying coefficients for a finite impulse response filter, the method comprising:

collecting sample input values for a function;

identifying desired output values for the filter for the collected sample input values;

approximating the output values from the input values using a linear fitting technique; and

setting the coefficients to values obtained from the linear-fitting technique.

44. The method of claim 43 wherein the results of the function are generated by calculating a working set size of a program image as the working set size is incrementally improved.

45. The method of claim 43 wherein the linear-fitting technique is for a certain-order finite impulse response filter and including determining an error between the approximated output values and the desired output values and when the difference between the determined error and input for the next higher-order finite impulse response filter is not significant using the certain-order finite impulse response filter.

46. The method of claim 45 wherein the significance of the difference between the determined error and input for the next higher-order finite impulse response filter is determined by correlating the determined error with the additional input for the next higher-order finite impulse response filter.

47. A method for converting a finite impulse response filter into an infinite impulse response filter so that a desired tradeoff between confidence and mean lag is established, the method comprising:

for each of a plurality of sets of autoregressive coefficients,

calculating results of the infinite impulse response filter with the set of autoregressive coefficients, and

generating a value that rates the confidence and mean lag for the calculated results; and

selecting the set of autoregressive coefficients with the generated value with the highest rating of confidence and mean lag for the infinite impulse response filter.

48. The method of claim 47 wherein the filters are for determining a rate of improvement in a working set optimization process.

49. The method of claim 47 wherein the plurality of sets are generated using Brent's method.

50. The method of claim 47 wherein the plurality of sets are generated using Powell's method.

51. A computer-readable medium containing instructions for causing a computer system to evaluate locality of references for a layout of a computer program, the computer program to be executed by a computer system with a page architecture, each page having memory locations, by:

for each of a plurality of selected memory locations of a page, estimating a working set size of the layout when the layout is positioned to start at that memory location; and

combining the estimated working set size as an indication of the locality of references of the layout of the computer program.

52. The computer-readable medium of claim 51 wherein the plurality of selected memory locations include each memory location of a page.

53. The computer-readable medium of claim 51 wherein the plurality of selected memory locations include less than all the memory locations on a page.

54. The computer-readable medium of claim 51 wherein the plurality of selected memory locations are separated by relatively prime number of memory locations.

55. The computer-readable medium of claim 51 wherein the plurality of selected memory locations includes less than all the memory

locations on a page and are approximately evenly distributed throughout the page.

56. The computer-readable medium of claim 51 wherein the plurality of selected memory locations includes less than all the memory locations on a page, are approximately evenly distributed throughout the page, and are separated by relatively prime number of memory locations.

57. The computer-readable medium of claim 51 wherein the combining of the estimated working set size includes averaging the estimated working set sizes.

58. A computer system for estimating the rate of improvement in the working set size for a plurality of layouts of a computer program, the layouts resulting from a layout optimization process, comprising:

a first estimating component that estimates the change in working set size from one improved layout to the next improved layout;

a second estimating component that estimates the time needed to improve the layout; and

a combining component that combines the estimated change in working set size with the estimated time needed to improve the working set for that layout to estimate the rate of improvement.

59. The system of claim 58 wherein the estimating of the change in working set size includes filtering the actual change in working set size from one layout to the next.

60. The system of claim 58 wherein the estimating of the time to improve the layout includes evaluating a formula that estimates the time based on

estimated numbers of times that various sub-steps are performed when improving the layout.

61. The system of claim 60 wherein the estimated numbers of times that various sub-steps are performed are based on evaluating actual results of the improvement.

62. The system of claim 58 wherein the estimating of the change includes filtering the actual change in working set size from one layout to the next and wherein the estimating of the time to improve the layout includes evaluating a formula that estimates the time based on number of times various sub-steps are performed when improving the layout.

63. The system of claim 58 wherein the estimating of the change in working set size filters an actual change in working set size and the estimating of time computes the time based on number of times the layout has been improved.

64. The system of claim 58 including terminating the system when the estimated rate of change per time period is outside of a threshold rate of change.

65. The system of claim 58 wherein the estimating of the change in the working set size includes filtering the actual change in working set size from one layout to the next with a filter that is generated from analysis of working set size produced during the improvement of other layouts.

66. The system of claim 65 wherein the filter is generated by using a frequency-domain analysis of an actual rate of change per improvement

and defined rate of change per improvement, the defined rate of change being calculated for each improvement based on knowledge of working set size of subsequent improvements.

67. The system of claim 65 wherein the filter is generated by using a time-domain analysis of an actual rate of change per improvement and defined rate of change per improvement, the defined rate of change being calculated for each improvement based on knowledge of working set size of subsequent improvements.

68. A computer-readable medium containing instructions for causing a computer system to convert a finite impulse response filter into an infinite impulse response filter so that a desired tradeoff between confidence and mean lag is established, by:

for each of a plurality of sets of autoregressive coefficients, for the infinite impulse response filter calculating results of the infinite impulse response filter with the set of autoregressive coefficients; and

selecting the set of autoregressive coefficients with a highest rating of the trade off between confidence and mean lag for the infinite impulse response filter.

69. The computer-readable medium of claim 68 wherein the filters are for determining a rate of improvement in a working set optimization process.

70. The computer-readable medium of claim 68 wherein the plurality of sets are generated using Brent's method.

71. The computer-readable medium of claim 68 wherein the plurality of sets are generated using Powell's method.